

МАКСИМ МИХАЙЛОВИЧ ЧАЛЫШЕВ

# ORACLE SQL. 100 ШАГОВ ОТ НОВИЧКА ДО ПРОФЕССИОНАЛА

20 ДНЕЙ НОВЫХ ЗНАНИЙ И ПРАКТИКИ

**Максим Михайлович Чалышев**  
**Oracle SQL. 100 шагов**  
**от новичка до профессионала.**  
**20 дней новых**  
**знаний и практики**

*[http://www.litres.ru/pages/biblio\\_book/?art=43200700](http://www.litres.ru/pages/biblio_book/?art=43200700)*  
*ISBN 9785005006189*

**Аннотация**

Более 300 ответов на вопросы. Более 500 практических заданий. Более 1000 разобранных примеров. Учебник справочник по языку SQL.

# Содержание

Введение	8
День первый	12
Шаг 1. Что такое SQL, назначение языка	13
Что такое базы данных, назначение баз данных	15
Веб-технологии	16
Мобильные устройства	17
Игры	18
Крупные корпорации	19
Назначение языка SQL, необходимость изучения этого языка	20
Вопросы учеников	22
Шаг 2. Теория и практика. Учебная схема данных. Организация работы	23
Общая схема процесса обучения, или Как читать данную книгу	25
Несколько терминов	26
Учебная схема	27
Вопросы учеников	32
Шаг 3. Подготовка к работе. Процесс обучения.	33
Описание интерфейса ORACLE APEX	
Подготовка к работе	33
Процесс обучения	34

Составьте карточки	37
Составляйте свой список вопросов	38
Создайте свое задание	39
Интерфейс ORACLE APEX	40
Просмотр структуры таблиц	42
Просмотр кода процедур	44
Вопросы учеников	45
Контрольные вопросы и задания для самостоятельного выполнения	46
Шаг 4. Таблицы в базе данных	47
Теория	47
Таблицы нашей учебной схемы	50
Вопросы учеников	53
Контрольные вопросы и задания для самостоятельного выполнения	54
Шаг 5. Типы данных	55
Теория и практика	56
Важные замечания	59
Вопросы учеников	60
Контрольные вопросы и задания для самостоятельного выполнения	62
День второй	63
Шаг 6. Создание таблиц	64
Важные замечания	65
Теория и практика	66
Вопросы учеников	69

Примеры	70
Примеры	71
Контрольные вопросы и задания для самостоятельного выполнения	72
Конец ознакомительного фрагмента.	73

**Oracle SQL.**  
**100 шагов от новичка**  
**до профессионала**  
**20 дней новых**  
**знаний и практики**

**Максим Михайлович**  
**Чалышев**

*SQL. 100 шагов от новичка до профессионала*

© Максим Михайлович Чалышев, 2019

ISBN 978-5-0050-0618-9

Создано в интеллектуальной издательской системе Ridero

20 дней новых знаний и практики.  
Более 300 ответов на вопросы.  
Более 500 практических заданий.  
Более 1000 разобранных примеров.

*Данную книгу я посвящаю своим друзьям:*

*Кузнецову Алексею – профессионалу управления  
в сфере ИТ,*

*Коршакову Артему – будущему высококлассному ИТ-  
специалисту.*

*Чальшиев Максим*

# Введение

Приветствую. Сначала как автор этой книги расскажу немного о своем профессиональном опыте. На данный момент я работаю в сфере информационных технологий уже почти 20 лет.

Основной моей специализацией в ИТ были и остаются базы данных и, прежде всего, СУБД ORACLE.

В первый раз я познакомился с данной СУБД в институте, один из моих преподавателей проходил стажировку в США. Он рассказывал студентам о базе данных ORACLE, о применении на производстве, в финансовых организациях, в крупных государственных учреждениях.

Во время изучения я был очарован четкой и понятной организацией структур информации в СУБД ORACLE, обширностью средств работы с данными, ее мощностью и вместе с тем неповторимой гибкостью, а также уникальными возможностями данной СУБД.

Далее, после института и защиты диплома, в течение нескольких лет я работал на крупном производственном предприятии, информационная система которого была построена преимущественно на использовании СУБД



# ORACLE.

Потоки данных представлялись грандиозными для того времени, каждый день сервер ORACLE рассчитывал сотни тысяч производственных компонентов, технологических маршрутов, спецификаций изделий.

Стоит понимать, что на дворе был конец XX века и объемы, которые тогда казались нам огромными, сейчас вызывают улыбку, технологические мощности ушли далеко вперед, и сейчас даже средний ноутбук сопоставим по быстродействию с многопроцессорными серверами тех лет.

Поле этого я перешел в сектор телекоммуникаций, и именно здесь в своей работе я столкнулся с быстрыми транзакциями, а также обработкой огромных объемов данных в сотни и сотни миллионов записей.

Далее была длительная работа в процессинге крупного банка, где также использовалась СУБД ORACLE, интернет-сатрапе средней компании-разработчика ПО, на этом этапе я познакомился с СУБД других производителей, таких как MS SQL PostgreSQL, MySQL; также я работал на большом DWH-проекте, где объемы в миллиарды записей передавались за считанные минуты.

И каждый раз, сталкиваясь с новой для себя сферой, я открывал дополнительные возможности и новые уникальные приемы разработки, которыми и хотел бы поделиться в этой книге.

Именно после многих лет работы начинаешь осознавать, что практика и теория отличаются. Те примеры, которые описаны в документации, могут работать по-разному в разных условиях, а также иметь множество нюансов использования в конкретной ситуации в конкретной сфере.

Основой моей работы была именно СУБД ORACLE, хотя я также неизбежно сталкивался вплотную и с другими технологиями, такими как Java, SAS, Python, веб-разработка JavaScript, Node JS.

Важно осознавать, что ORACLE сейчас представляет собой целый конгломерат производственных решений, куда входит, например, ORACLE Siebel CRM, JaVA, ORACLE Service Bus, но все же основным продуктом данной корпорации была и остается именно СУБД ORACLE.

Что же касается настоящего момента, то сейчас я провожу специальные курсы по SQL в базовой и расширенной версиях, на которые каждый из вас может записаться.

Адрес курсов [www.sqladv.ru](http://www.sqladv.ru)

Вопросы учеников, часть практических примеров взяты непосредственно с этих курсов.

# День первый

# Шаг 1. Что такое SQL, назначение языка

Приветствую вас, уважаемый читатель. Позволю написать несколько слов о себе.

На текущий момент вот уже более 20 лет я работаю IT-специалистом. Я занимал должности архитектора, администратора баз данных, разработчика баз данных.

У меня также есть своя IT-школа [sqladv.ru](http://sqladv.ru), где один из курсов, который я веду сам, посвящен разработке баз данных и языку SQL.

Эта книга представляет собой также своеобразный курс обучения: с помощью данной книги, упорно занимаясь, вы освоите язык SQL от начального уровня до уровня ведущего разработчика.

В данной книге рассмотрен диалект языка ORACLE SQL как один из наиболее распространенных на сегодняшний день. Также специалисты ORACLE обычно имеют более высокую зарплату по сравнению с другими разработчиками.

Когда я преподавал в своей IT-школе [sqladv.ru](http://sqladv.ru), то убедил-

ся, что программирование – это, прежде всего, практика, поэтому в каждой главе данной книги разбираются актуальные примеры и всегда присутствует несколько практических заданий, обязательных для выполнения.

В каждой главе книги вы сможете найти наиболее интересные и актуальные вопросы моих учеников, разумеется, с моими ответами.

Итак, перейдем непосредственно к теоретической части.

# **Что такое базы данных, назначение баз данных**

Трудно себе представить, что раньше вся информация размещалась на бумажных носителях и архивы документов занимали подчас целые здания.

Сейчас же пришел новый век, новая информационная эпоха, и теперь для хранения и обработки информации используются в основном электронные системы.

Огромное количество таких систем работает под управлением баз данных. Спектр применения систем управления базами данных на сегодняшний день практически необъятен — базы данных используются в интернете, в производстве, в промышленности, в маркетинге, в мобильных устройствах, в финансовой и банковской сферах, на телевидении, в телекоммуникациях и рекламе.

Какова применимость баз данных, то есть где используются базы данных?

Вот лишь некоторые области, где базы данных нашли применение.

# Веб-технологии

Веб-технологии проникли в нашу жизнь, и без них уже сложно представить современный мир. Социальные сети, почта, поисковые системы, онлайн-сервисы погоды и навигации – всем этим большинство из нас пользуется ежедневно.

Почти все онлайн-ресурсы работают с базами данных. Практически каждый сайт, поисковая система, социальная сеть построены на основе баз данных и используют язык SQL.

Поэтому, если ваш бизнес или ваша профессия каким-либо образом связаны с интернет-проектами, если вы веб-дизайнер или веб-программист (и не важно, на каком языке вы специализируетесь), знание баз данных и языка SQL вам обязательно пригодится в работе.



# Мобильные устройства

Большинство мобильных приложений, приложений для планшетов также использует базы данных в своей работе. Система управления базами данных для мобильных устройств называется SQLite. SQLite имеет ряд особенностей, связанных с характеристиками мобильных устройств, но в целом использует такой же синтаксис SQL, как и другие базы данных.

# Игры

Современные компьютерные игры также невозможны без использования баз данных. Игры – это множество объектов, игроков, карт, вооружений, стратегий, юнитов... С этой информацией надо активно работать. Без систем управления базами данных тут не обойтись. И если вы планируете связать свою жизнь с интереснейшей профессией игрового дизайнера или программиста компьютерных игр, знание баз данных будет для вас очень и очень желательным.

# Крупные корпорации

Во всех без исключения крупных компаниях используются базы данных. Системы управления базами данных управляют банкоматами и пунктами выдачи наличных, на них реализованы бухгалтерия, учет и управление кадрами. Системы управления базами данных считают телефонные звонки по тарифам, вычисляют стоимость интернет-подключений и трафика.

Если вы работаете в крупной компании или собираетесь связать свою карьеру с работой на корпорацию, вам необходимо понимать, что такое база данных, принципы ее устройства, знать и понимать язык для работы с базами – SQL.

# **Назначение языка SQL, необходимость изучения этого языка**

Structured Query Language (SQL) – язык структурированных запросов.

Язык запросов SQL – универсальный язык для работы с данными базы. Язык запросов SQL используется для управления массивами данных в БД, множествами.

Язык SQL предоставляет возможность для вывода структурированной заданной информации из базы. SQL также применяется для изменения данных, добавления данных из базы.

Язык SQL относится к функциональным языкам программирования. Он отличается от алгоритмических языков. Основу языка составляет не алгоритм как таковой, а совокупность команд, определяющих взаимоотношения информационных множеств и подмножеств.

Следует отметить, что системы управления базами данных – СУБД – имеют различные реализации, такие как ORACLE, MS SQL, MY SQL.

Язык SQL в разных СУБД имеет небольшие отличия, например в детальном синтаксисе описания операторов.

Такие отличия присутствуют в специальных функциях, относящихся к той или иной СУБД, но все же в основном язык – это общий синтаксис, практически идентичный для любой СУБД.

В данном курсе мы будем рассматривать общепринятый синтаксис SQL ORACLE.

Данная книга, как я ранее писал, обучает диалекту ORACLE SQL как наиболее востребованному и сложному, но на страницах книги вы также сможете найти некоторые примеры на других диалектах.

# Вопросы учеников

*Я программирую на PHP, пригодятся ли мне знания из данной книги?*

Да, язык PHP используется для доступа к данным команды SQL, поэтому если вы намерены повышать свой профессиональный уровень, вам необходимо изучить материалы этой книги.

*Какой уровень знаний у меня будет после прочтения данной книги и выполнения всех практических заданий?*

Вы будете знать SQL на профессиональном уровне, вполне достаточном для разработки сложных баз данных.

*В данный момент получили широкое распространение NoSQL базы данных, какой язык используется для работы с такими СУБД?*

Для каждой NoSQL СУБД используется свой язык программирования, отличный от SQL, разумеется.

*С каким уровнем знаний можно приступить к чтению этой книги?*

С начальным уровнем знаний. Вполне достаточно небольшого уровня компьютерной грамотности.

# **Шаг 2. Теория и практика.**

## **Учебная схема данных.**

### **Организация работы**

Научиться языку SQL на профессиональном уровне по данной книге вполне реально, более того – я не вижу причин, по которым это может не получиться. Разумеется, многое зависит от вашего личного упорства, методичности обучения, системности выполнения практических заданий. Выделите каждый день по три—четыре часа вашего времени, и уже через три недели вы сможете писать SQL-запросы любой сложности.

Книга называется «100 шагов», и это соответствует действительности: это 100 шагов, которые вам необходимо пройти, для того чтобы овладеть SQL и базами данных на профессиональном уровне. Каждый шаг представляет собой отдельную главу книги. Глава книги – обзор определенного вопроса или темы по предмету: базы данных или язык SQL.

Каждая глава поделена, в свою очередь, на следующие разделы:

1. **Введение** – в этом разделе рассказывается, собственно, о предмете или теме, которой посвящается данная глава книги.

2. **Теория и практика** – читателю даются теоретические обоснования темы, разбирается синтаксис рассматриваемых в главе операторов. Приводятся понятные и доступные примеры.

3. **Важные замечания** – в любой теме есть свои особенности, свои нюансы, эти нюансы и обобщаются в разделе.

4. **Вопросы учеников** – здесь я отвечаю на наиболее популярные и интересные вопросы, которые задавали мне слушатели моих курсов.

5. **Контрольные вопросы и задания для самостоятельного выполнения** – вопросы по уроку, вопросы и задания, которые вам необходимо решить самим.



# **Общая схема процесса обучения, или Как читать данную книгу**

Книгу следует рассматривать как учебное пособие, и я не стану скрывать, что в книге используются материалы моих курсов по обучению базам данных и языку SQL школы [sqladv.ru](http://sqladv.ru).

Внимательно изучите теорию, запомните, проанализируйте синтаксис команд.

Все примеры необходимо прорешивать, самостоятельно писать запросы, создавать схемы, размышлять над теоретическим обоснованием примера, то есть стараться понять, почему получается так, а не иначе.

Практические задания также необходимо решать самостоятельно, и при решении данных заданий следует обратиться к материалам из теории.

Особое внимание следует уделить вопросам, рассматриваемым в разделах «Важные замечания» и «Вопросы учеников». Именно в этих разделах, как правило, содержится наиболее важная и полезная информация.

# Несколько терминов

Иногда в тексте книги могут встретиться некоторые сокращения и специальные термины; расшифруем их значение.

СУБД – система управления базами данных – совокупность программных средств, обеспечивающих управление созданием и использованием баз данных. Наиболее распространенные: ORACLE, MS SQL, mySQL, PostgreSQL.

БД – база данных – совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

# Учебная схема

Для занятий нам понадобится учебная схема данных.

И есть хорошая новость.

Многие примеры начального курса могут быть выполнены онлайн, для этого вам потребуется просто перейти по заданной ссылке и заполнить некоторые поля.

Итак, большинство примеров и практических заданий вы сможете выполнить на онлайн-сервисах.

Первый сервис предоставляет компания ORACLE для разработки, я уже зарегистрировал там аккаунт, и сейчас там уже есть все таблицы, которые нам потребуются в учебе.

Также там есть все необходимые заполненные данные для выполнения учебных задач и запросов.

Вам достаточно пройти по ссылке

<https://apex.oracle.com/pls/apex/>

и ЗАПОЛНИТЬ регистрационную информацию.



## Oracle Application Express



sqladv



student1



●●●●●●●●



Remember workspace and username



**Sign In**

[Reset Password](#)

[Request a Workspace](#)

Deutsch · **English** · Español · Français · Italiano · Português (Brasil) ·  
中文 (繁體) · 中文 (简体) · 日本語 · 한국어

Рисунок 1. Форма авторизации в APPEX

Первое поле сверху мы заполняем SQLADV, во второе поле мы вносим имя пользователя student1 и заполняем пароль – также student1.

Также будут работать учетные записи: student2/, student2, student3/, student3... student11/, student11).

Перед вами откроется среда разработки.

Выберите пункт меню SQL Workshop, а дальше SQL ComMANd.

Перед вами откроется среда выполнения SQL-запросов.

Напишите следующий учебный запрос:

**SELECT \* FROM AUTO;**

Нажмите кнопку RUN SQL, и далее в нижней части экрана должен появиться результат выполнения запроса.

Это и будет тот сервис, которым вы в основном сможете пользоваться для выполнения практических заданий.

Данный ресурс предоставлен компанией ORACLE в рекламных маркетинговых целях.

Если не устраивает данный сервис или почему-то этот сервис у вас не работает, тогда существует второй способ.

Вы можете воспользоваться сервисом SQL-Фидель.

Учебная схема, по которой необходимо заниматься, состоит из нескольких таблиц, которые заполнены соответствующими данными.

ющими данными.

Введите в поле браузера ссылку

<http://sqlfIDdle.com/>.

Выберите тип базы данных ORACLE 11 g r2.

Далее вам потребуется загрузить учебную схему, по которой мы будем заниматься. Скрипт учебной схемы находится по следующей ссылке в текстовом документе:

<http://sqladv.ru/dev/sql.txt>.

Скопируйте содержимое скрипта в поле в левой части экрана и нажмите кнопку BuildSchema.

После чего уже в правой части экрана SQLFIDdle вы сможете писать необходимые запросы и выполнять учебные задания.

Учебные запросы в SQLFIDdle пишутся в текстовом поле в правой части экрана.

После создания схемы напишите следующий учебный запрос:

**SELECT \* FROM MAN**

Нажмите RUN SQL, в нижней части экрана должен появиться результат выполнения запроса.

The screenshot shows the SQL Fiddle interface. The top bar includes the URL 'sqlfiddle.com/4465626/1', the database 'Oracle 11g R2', and buttons for 'View Sample Fiddle', 'Clear', and 'Text to DDL'. The main area is split into two panes. The left pane contains SQL code for creating a table and inserting data. The right pane shows a SQL query. Below the panes are buttons for 'Build Schema', 'Edit Fullscreen', 'Browser', 'Run SQL', 'Edit Fullscreen', and a zoom icon. At the bottom, a table displays the data inserted into the 'MAN' table.

```
1104 /
1105
1106 INSERT INTO MAN (PHONENUM, FIRSTNAME, LASTNAME, CITYCODE, YEAROLD)
1107 VALUES('9213333336', 'Алексей', 'Данасов', 7, 25)
1108 /
1109
1110 INSERT INTO MAN (PHONENUM, FIRSTNAME, LASTNAME, CITYCODE, YEAROLD)
1111 VALUES('9213333331', 'Андрей', 'Николаев', 2, 27)
1112 /
1113
1114 INSERT INTO MAN (PHONENUM, FIRSTNAME, LASTNAME, CITYCODE, YEAROLD)
1115 VALUES('9213333332', 'Макс', 'Моситов', 2, 21)
1116 /
1117
1118 INSERT INTO MAN (PHONENUM, FIRSTNAME, LASTNAME, CITYCODE, YEAROLD)
1119 VALUES('9214444444', 'Алиса', 'Нихафо', 3, 38)
1120 /
1121
```

```
1 select * from MAN
```

PHONENUM	FIRSTNAME	LASTNAME	CITYCODE
9152222221	Андрей	Николаев	1
9152222222	Максим	Моситов	1
9153333333	Олег	Данасов	3
9173333334	Алиса	Нихафо	4

## Рисунок 2. Пробный запрос SQLFiddle

Если все получилось, то вы можете приступить к учебе.

# Вопросы учеников

*Так все-таки в каком из сервисов лучше выполнять практические задания?*

В любом, лично мне больше нравится сервис APEX.

*Я пытаюсь залогиниться в сервис APEX, но появляется сообщение об ошибке. В чем может быть дело?*

Проверьте правильность ввода имени пользователя и пароля. Самое верхнее поле должно содержать значение SQLADV.

*Во время работы с сервисом SQLFIDlle при создании схемы возникает ошибка.*

Проверьте, пожалуйста, правильно ли установлен переключатель СУБД – значение выпадающего списка должно быть ORACLE 11g. Также проверьте, скопирован ли скрипт при создании схемы.



# **Шаг 3. Подготовка к работе. Процесс обучения. Описание интерфейса ORACLE APEX**

## **Подготовка к работе**

До шага 51 все занятия и практические упражнения можно выполнить с использованием онлайн-сервисов, в шаге 51 подробно описано, какое программное обеспечение следует установить дополнительно и как это сделать.

# Процесс обучения

Как я уже говорил, самое важное в процессе обучения языку SQL – это именно практика, при этом не важно, как много вы знаете. Главное – научиться использовать свои знания в работе.

Тысяча практических задач из данной книги, прорешенных вами, даст гораздо больший эффект, чем изучение одной лишь теории.

Данная книга ориентирована исключительно на практическую работу. Теория здесь поясняется практическими примерами.

Изучив теорию и синтаксис выражения, сразу приступайте к практике, чтобы понять, как работает данный синтаксис, данное выражение.

Практические задания находятся сразу после пояснения теоретических положений. Примерный текст практического задания выглядит следующим образом:

Выберите из таблицы автомобилей (AUTO) машины синего и зеленого цветов.

Попробуйте сначала самостоятельно написать запрос. Сравните свое решение с решением, приведенным в книге.

Если сразу не получается, тогда внимательно изучите решение, приведенное в книге, и постарайтесь понять сам принцип, как решается данная задача, что является главным в решении.

После решения каждого из заданий переходите к заданиям для самостоятельного выполнения.

В данной книге рассматривается более тысячи практических примеров и упражнений по SQL.

Кроме того, в книге есть не только практические задачи – после примеров в некоторых шагах встречаются контрольные вопросы по теоретическим материалам.

Такие вопросы необходимы для более четкого понимания изучаемой темы.

Внимательно изучите вопрос и постарайтесь дать на каждый вопрос развернутый и подробный ответ.

Некоторые названия таблиц заменены их смысловыми обозначениями. Вам следует самостоятельно понять, исходя из смысла слова, в какой таблице находятся соответствующие данные.

На самом деле основных таблиц всего три, так что это будет несложно. Например, если речь идет об авто, тогда это таблица AUTO, если речь идет о людях, покупателях, тогда имеется в виду таблица MAN, если о городах – CITY.

Вот несколько практических советов, как улучшить процесс обучения SQL.

# Составьте карточки

Наиболее сложные, трудно запоминаемые теоретические вопросы и ответы лучше записывать на карточки, чтобы в конце занятий повторять сложные темы.

Также рекомендуется повторять эти карточки через каждые 10 глав книги.

Данная карточка может выглядеть следующим образом. С одной стороны пишется ключевой вопрос: «*Какой оператор в SQL-запросах отвечает за группировку данных?*»

С другой стороны ответ: «*Оператор GROUP BY*».

И далее несколько примеров запросов с использованием данного оператора.

# Составляйте свой список вопросов

Для себя составьте дополнительные вопросы и ответьте на них.

Например, мы изучили агрегатные функции SUM, MAX, MIN, а как работает агрегатная функция COUNT? Не бойтесь задавать себе сложные вопросы и изучать новую информацию.

# Создайте свое задание

Придумайте свои практические задания и порешайте их.

В дополнение к практическим заданиям из книги продумайте свои собственные и попробуйте их решить.

Например, у вас есть практическое задание: выбрать из таблицы AUTO все автомобили марки BMW синего цвета.

Выбрать из таблицы AUTO все автомобили марки BMW синего и зеленого цветов.

# Интерфейс ORACLE APEX

Онлайн-сервис ORACLE APEX обладает рядом дополнительных возможностей, которые будут нам помогать в процессе обучения. Поэтому, если вы используете для выполнения практических задач APEX, рекомендую ознакомиться с данным материалом.

К дополнительным возможностям сервиса APEX относится просмотр объектов схемы данных.

После входа в сервис APEX следует воспользоваться пунктом меню SQL Workshop, выберите подпункт меню OBJECT BROWSER.

В левой части экрана располагается список объектов, где через выпадающий список указывается, объекты какого типа отражаются в списке.



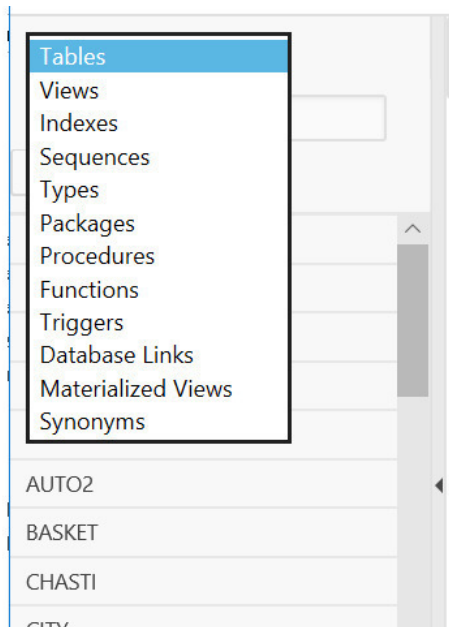


Рисунок 4. Список объектов

Выбор соответствующего типа объекта покажет список объектов заданного типа. Щелчок по заданному объекту позволяет отобразить структуру и свойства заданного объекта.

# Просмотр структуры таблиц

Выберите в выпадающем списке в левой верхней части формы TABLES. Список отразит все таблицы, которые вам доступны.

Выберите любую из таблиц.

В правой части страницы отразится структура выбранной таблицы.

Tables

Q

Tables

AUTO

AUTO1

AUTO11

AUTO2

BASKET

CHASTI

CITY

CL\_PHONES

DASACUCAK

DEMO1

AUTO

Table

Data

Indexes

Model

Constraints

Grants

Statistics

UI Defaults

Triggers

Add Column

Modify Column

Rename Column

Drop Column

Rename

Copy

Drop

Column Name	Data Type	Nullable
REGNUM	VARCHAR2(15)	No
MARK	VARCHAR2(10)	Yes
COLOR	VARCHAR2(15)	Yes
RELEASEDT	DATE	Yes
PHONENUM	VARCHAR2(15)	Yes

Download

Print

Также доступны следующие вкладки, которые показывают сведения о таблице:

- TABLE – структура избранной таблицы;
- Data – данные избранной таблицы;
- INdexes – сведения об индексах заданной таблицы;

- ConstraiNts – ограничения заданной таблицы;
- Grants – права базы данных по заданной таблице;
- SQL – SQL-код таблицы. Если вам необходимо посмотреть SQL-код таблицы, тогда следует обратиться к этой вкладке.

# Просмотр кода процедур

В некоторых шагах мы обращаемся к исходному коду процедур и функций.

Выберите в выпадающем списке одно из следующих наименований: SEQUENCEs, Function, Procedures, Packages.

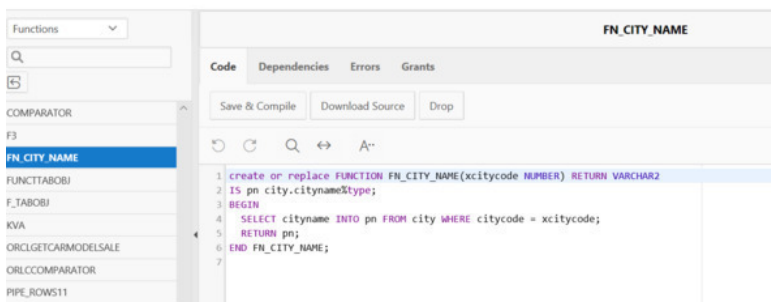


Рисунок 5. Исходный код функции Fn\_CITY\_NAME

В правой части веб-страницы будет отображен исходный код выбранного объекта.

# Вопросы учеников

*Сколько времени следует уделять занятиям?*

Рекомендую уделять занятиям не менее четырех часов в день, тогда за 20 дней вы успеете пройти все шаги.

*Если я все же не могу найти ответ на вопрос или не могу решить задание, что мне делать?*

На сайте [sqladv.ru](http://sqladv.ru) есть ссылка на нашу группу в «Фейсбуке», там вы наверняка найдете ответ и решение задачи, с которой испытываете трудности.

*В SQLFIDele есть такие же возможности по просмотру и редактированию таблиц, как в ORACLE APEX?*

Нет, *SQLFIDele* – это менее сложный инструмент, тем не менее его возможностей достаточно, чтобы выполнить большинство практических заданий из этой книги.

*Сколько примерно времени в пропорции уделять теории, а сколько посвятить практике?*

Лучше всего из четырех часов рабочего времени следует один час уделить теории, а три часа – практике. Таким образом, примерно 80 процентов вашего учебного времени должно занимать выполнение практических задач.

# **Контрольные вопросы и задания для самостоятельного выполнения**

1. Сколько рекомендуется тратить времени на занятия?
2. Как сделать карточки с наиболее сложными для понимания вопросами?
3. Как в APЕХ посмотреть все таблицы схемы?
4. Как в APЕХ посмотреть структуру заданной таблицы?

# Шаг 4. Таблицы в базе данных

## Теория

База данных – это, прежде всего, таблицы. Таблицы базы данных можно представить как таблицы в WORD или EXCEL, где в каждой ячейке содержатся определенные данные, но также есть и некоторые отличия.

Дело в том, что таблицы в базах данных создаются по некоторым правилам, и вот основные правила для таблиц в базе данных.

Так как мы изучаем SQL-диалект ORACLE СУБД, то данные правила справедливы именно для СУБД ORACLE:

- любая таблица в базе имеет уникальное наименование в рамках схемы данных;
  - у каждой таблицы всегда есть заданное количество колонок: больше нуля и меньше 1024;
  - каждая колонка также должна иметь уникальное наименование, но уже в рамках данной таблицы;
  - в таблице в базе данных может быть практически неограниченное количество строк, здесь ограничения касаются только объема диска базы данных;
  - для данных в таблице можно создавать ограничения.
- Ограничения касаются всех данных в колонке, на которую

установлено ограничение;

- имена таблиц, имена колонок имеют ограничения по количеству символов и не могут называться зарезервированным словом, например командой из языка SQL или PL SQL. Также наименование колонки таблицы не должно начинаться с цифр;
- имя колонки в рамках таблицы также должно быть уникальным.

Создание таблиц по указанным правилам – это первый шаг в разработке базы данных.

Таблицы в базе, состав колонок таблицы должны производиться в соответствии со стандартами проектирования реляционной базы данных.

Работа со структурой таблиц, данными в таблицах осуществляется с помощью языка запросов SQL.

**Одна или несколько колонок в таблице могут быть обозначены как первичный ключ.**

Первичным ключом обозначаются колонки таблицы, содержащие набор уникальных значений, по которым мы можем однозначно идентифицировать строку в рамках этой таблицы. Первичный ключ не может содержать пустые значения, так как всегда имеет ограничение NOT NULL.



**Вторичный ключ – так обозначается колонка таблицы, в которой есть данные, используемые для связи с другой таблицей.**

# Таблицы нашей учебной схемы

Наша учебная схема очень проста и состоит всего лишь из четырех таблиц.

Первая таблица MAN содержит сведения о людях, которые приобрели машины.

Колонки таблицы MAN:

- PHONEnum – уникальный телефонный номер человека, первичный ключ для таблицы MAN, содержит текстовые данные;
- CITYCode – код города, вторичный ключ для связи с таблицей CITY;
- FirstName – имя человека (текстовые данные);
- LAsTName – фамилия человека (текстовые данные);
- YearOld – возраст человека (числовые данные).

Таблица CITY – справочник городов, состоит из трех колонок:

- CITYCODE – уникальный код города, ключевое поле для таблицы CITY (числовые данные);
- CITYNAME – наименование города (текстовые данные);
- PEOPLES – население города, количество человек, которые проживают в городе (числовые данные).

Таблица AUTO – сведения об автомобилях автосалона.

Колонки таблицы AUTO:

- REGnum – уникальный регистрационный номер автомобиля (содержит текстовые данные);
- PHONEnum – телефонный номер покупателя, вторичный ключ для связи с таблицей MAN;
- MARK – марка авто (текстовые данные);
- COLOR – цвет авто (текстовые данные);
- ReleASeDT – дата создания авто, дата/время (специальный тип данных).

Таблица AUTO1 является копией таблицы AUTO и имеет те же колонки, что и таблица AUTO, и достаточно похожие данные, эта таблица используется в нескольких учебных заданиях (так же, как CITY1, MAN1).

Следующее изображение показывает основные таблицы в учебной базе данных в виде схемы:

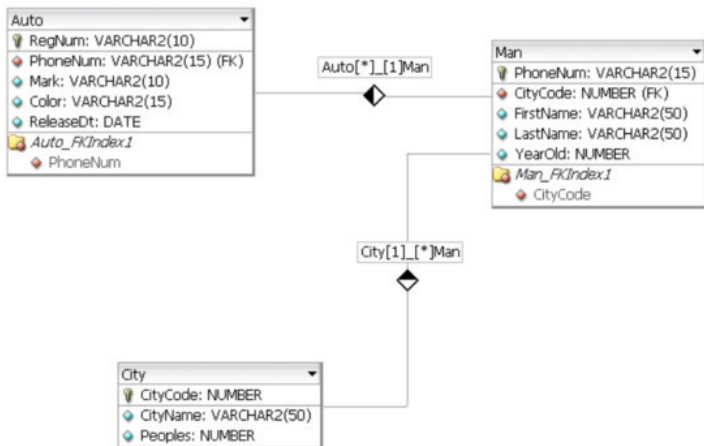


Рисунок 6. Учебная схема базы данных

# Вопросы учеников

*Вы сказали, что наименования таблиц должны быть уникальны в рамках одной схемы. Что такое схема?*

В СУБД есть понятие схемы – это особая логическая область, ассоциированная с заданной учетной записью, которая объединяет несколько объектов базы данных.

*Почему телефонный номер покупателя PHONEunit – текстовое поле, разве оно не должно быть числовым?*

Иногда телефонный номер заполняют со скобками. Чтобы разрешить это противоречие, я использовал текстовый (VARCHAR2) тип данных для этой колонки; кроме того, так сделать правильно, так как это упрощает поиск нужных нам номеров по префиксу.

*Какие команды SQL позволяют изменять структуру таблицы, добавлять новые колонки, например?*

Это команда ALTER TABLE, с которой мы познакомимся чуть позже.

# **Контрольные вопросы и задания для самостоятельного выполнения**

1. Могут ли колонки разных таблиц называться одинаково?
2. Что такое первичный ключ, какие колонки (первичные ключи) есть в нашей учебной схеме?
3. Какие данные находятся в учебной таблице CITY?
4. Какая колонка в таблице MAN нашей учебной схемы содержит данные о возрасте человека?

## Шаг 5. Типы данных

Для удобства в SQL все данные разделены на различные типы: например, есть строковый тип, к которому относятся только строки и текст; есть целочисленный тип, к нему можно отнести только целые числа; определен специальный тип данных для чисел с плавающей точкой.

Каждой колонке в таблице назначается свой определенный тип данных, то есть каждая колонка таблицы может сохранять данные строго заданного типа и никаких данных другого типа, отличного от этого. Например, в одной из колонок могут находиться только строки и текст, а в другой – только числа.

Типов данных в SQL ORACLE-диалекте множество, мы же рассмотрим самые основные из них.

# Теория и практика

Ниже приведена таблица основных типов данных, используемая в SQL ORACLE. В таблице колонка-размер означает, какой максимальный объем информации сможет вместить этот тип данных. Например, тип данных VARCHAR2 может вместить в строку длиной не более 4000 символов.



Типы данных	Размер	Описание
char(размер)	Максимальный размер 2000 байт.	<p>Для текстовых данных не превышающих 2000 символов. Статический выделение памяти под текст. Если сохраняемое значение короче, строка дополняется пробелами до 2000 символов.</p> <p>В скобках указывается количество символов фиксированной длины</p>
nvarchar2(размер)	Максимальный размер 4000 байт.	<p>Специальный тип для сохранения текстовых данных в формате UNICODE. В скобках указывается – количество сохраняемых символов в кодировке Unicode переменной длины.</p>
varchar2(размер)	Максимальный размер 4000 байт.	<p>Для строк и текста не превышающих 4000 символов.</p> <p>В скобках указывается количество сохраняемых символов переменной длины.</p>
CLOB	Максимальный размер 2GB.	<p>Для сохранения больших текстов.</p> <p>Символьные данные переменной длины.</p>
raw	Максимальный размер 2000 байт.	<p>Для двоичных данных переменной длины</p>
number(точность,масштаб)	<p>Точность может быть в диапазоне от 1 до 38.</p> <p>Масштаб может быть в диапазоне от -84 до 127.</p>	<p>Например, number (14,5) представляет собой число, которое имеет 9 знаков до запятой и 5 знаков после запятой.</p>

## таблица. Типы данных

	до 38.	и 2 знака после .
date	date может принимать значения от 1 января 4712 года до н.э. до 31 декабря 9999 года нашей эры.	Используется для хранения информации дата время.

## Таблица. Типы данных

# Важные замечания

Для удобства (во избежание излишней путаницы) в учебных примерах для этой книги рассматриваются в основном только три типа данных, однако нам достаточно этих типов для решения большинства учебных задач и понимания учебного материала.

Основные типы данных, используемые в книге в практических задачах:

- **VARCHAR2 (n)** – тип для хранения текстовой информации, в скобках указывается максимальное количество символов в строке. Данный тип используется при работе со строковыми данными разной длины, память под такие данные выделяется динамически;
- **NUMBER** – тип данных для хранения числовой информации, причем можно использовать как для целых чисел, так и для чисел с плавающей точкой;
- **DATE** – специальный тип данных для сохранения специальной информации – дата-время, например дата и время создания записи, дата и время электронной подписи документа, дата и время заключения сделки.

Эти типы данных достаточно часто используются на практике и применяются в работе.

# Вопросы учеников

*Вы рассказали про тип данных VARCHAR2 для хранения строк, но в виде строки можно записать и числа, и даты тоже. Зачем так много разных типов, может, они не нужны?*

Да, вы можете сохранять числа как строки, но в дальнейшем вам будет сложно работать с этими данными, обработка будет затруднена. Например, со строками нельзя производить математических вычислений, и вам придется применять операции преобразования, что негативно скажется на качестве и скорости работы вашей программы.

*Типы данных в других SQL СУБД, отличных от ORACLE, также различаются?*

Есть незначительные различия, например в MS SQL используется VARCHAR, а не VARCHAR2, или вместо CLOB используется тип TEXT. Необходимо обратиться к соответствующему разделу документации выбранной СУБД, чтобы понять, какие именно типы данных различаются.

*Почему в ORACLE SQL используется именно VARCHAR2, а не просто VARCHAR, как в MS SQL-сервер?*

В ORACLE SQL тоже существует тип VARCHAR, но исторически сложилось так, что в ORACLE SQL между этими

двумя типами существует разница:

- `VARCHAR` может хранить до 2000 символов, а `VARCHAR2` может хранить до 4000 символов;
- если мы объявим тип данных как `VARCHAR`, то будет зарезервировано место для пустых `NULL VALUES`.

Поэтому чаще всего на практике в `ORACLE` используется `VARCHAR2`.

# **Контрольные вопросы и задания для самостоятельного выполнения**

1. Для чего используется тип DATE?
2. Нам необходимо в одной из колонок таблиц сохранять целые числа, какой тип следует использовать?
3. Какой тип правильно использовать для хранения данных о цвете автомобиля?
4. Какой тип данных необходимо использовать для хранения сведений о росте человека?

# День второй

## Шаг 6. Создание таблиц

Как мы уже поняли, основная информация, с которой нам предстоит работать в базе данных, находится в таблицах. Из шага 4 (таблицы в базе данных) про таблицы мы узнали следующие важные сведения:

- каждая таблица обладает уникальным наименованием;
- у таблицы обязательно должны быть колонки, каждая из которых обладает уникальным наименованием в рамках этой таблицы;
- для каждой колонки таблицы задается свой тип данных (про типы данных подробно рассказывается в предыдущем шаге).



# Важные замечания

1. Имена таблиц, имена колонок в SQL имеют ограничение по количеству символов и не могут называться зарезервированным словом, таким как команда из языка SQL.

Например, нельзя назвать таблицу или колонку GROUP, это зарезервированное слово, часть команды GROUP BY, или же недопустимо называть таблицу или колонку таблицы FROM, SELECT, INSERT, KEY.

2. Имена таблиц и имена колонок не могут начинаться с цифр; также принято использовать при именовании таблиц и колонок латинские буквы, хотя, впрочем, вполне возможно назвать таблицы и колонки таблиц на русском, китайском и даже хинди (я видел и такое), но подобные выкрутасы не приветствуются.

3. Очень желательно, чтобы наименования таблиц, а также наименования колонок таблиц отражали смысл относительно данных, которые содержатся в этих таблицах, например: MANS, CARS, STAFF – люди, машины, персонал, GOODS – товары, ITEMS – элементы.

# Теория и практика

Теперь, когда мы разобрались с теоретической частью, самое время заняться практическими упражнениями.

Разберемся на практике, как создавать таблицы в базе, используя язык SQL.

Для создания таблиц используется специальная команда SQL CREATE TABLE.

Синтаксис упрощенный.

```
CREATE TABLE имя таблицы (  
    Column_NAME1 column_type (NUMBER, или  
    VARCHAR2 (n) или DATE) primary KEY,  
    Column_NAME 2 column_type (NUMBER, или  
    VARCHAR2 (n) или DATE),  
    Column_NAMEn column_type (NUMBER или  
    VARCHAR2 (n) или DATE)  
);
```

Для простоты на начальном этапе в наших практических примерах мы будем использовать три основных типа данных. (NUMBER, VARCHAR2 (n), DATE), соответственно, для хранения чисел, строковых данных и данных календарного типа (дата/время).

Для текстового типа VARCHAR2 (n) после VARCHAR2 в скобках указывается количество символов для данной колонки.

Итак, сначала идет команда создания таблицы CREATE TABLE, далее – наименование таблицы: MANS, GOODS, ITEMS или любое другое.

Далее в скобках через запятую перечисляются наименования колонок и тип колонок.

Вот несколько примеров, как создавать таблицы в языке SQL:

1. Создать таблицу «Мебель»:

- артикул;
- наименование;
- количество;
- номер партии.

```
CREATE TABLE furnit (artikl VARCHAR2  
(50) PRIMARY KEY, NAME VARCHAR2  
(50),partCOUNT NUMBER, partnum NUMBER);
```

2. Создать таблицу «Корзина для веб-магазина»:

- артикул;
- наименование товара;
- имя покупателя;
- количество;
- дата покупки.

```
CREATE TABLE shopINGcart (  
article VARCHAR2 (50) PRIMARY KEY  
,itemName VARCHAR2 (50)  
,buyerNAME VARCHAR2 (50)
```

```
,itemCOUNT NUMBER  
,dtbuy DATE  
);
```

Создать таблицу «Предприятие»:

- название бригады;
- номер бригады;
- количество человек;
- дата создания;
- направление деятельности.

```
CREATE TABLE plant  
(  
  NAMEteam VARCHAR2 (15),  
  numteam NUMBER PRIMARY KEY,  
  MANCOUNT NUMBER,  
  crDATE DATE,  
  dirToDo VARCHAR2 (30)  
);
```

# Вопросы учеников

*Можно ли использовать заглавные буквы в языке SQL и когда это допустимо?*

Язык SQL не зависит от регистра, то есть при составлении команд можно писать и заглавными, и строчными буквами.

# Примеры

**Create TABLE Tab1 (TABNo INteger PRIMARY KEY, NAME VARCHAR2 (10));**

**Create TABLE Tab1 (TABNo INteger PRIMARY KEY, NAME VARCHAR2 (10));**

**CREATE TABLE Tab1 (TABNo INTEGER PRIMARY KEY, NAME VARCHAR2 (10));**

*Как переносить команды SQL на другую строку, если в одну строчку не помещается, существуют ли какие-то специальные правила?*

Язык SQL допускает достаточно вольный перенос строк, главное, не разделять этим переносом осмысленные команды, а также соблюдать последовательность команд.

# Примеры

Можно написать так:

```
CREATE TABLE TAB1 (TABno INteger  
PRIMARY KEY, NAME VARCHAR2 (10));
```

А можно и так:

```
CREATE TABLE  
TAB1 (  
TABno INteger PRIMARY KEY,  
NAME VARCHAR2 (10));
```

А вот такая запись уже неверна:

```
CREATE TABLE TAB1 (TABno INteger  
PRIMARY  
KEY, NAME VARCHAR2  
(10));
```

Еще один пример неверной записи:

```
CREATE TABLE  
PRIMARY KEY  
TAB1 (TABno INteger,  
NAME VARCHAR2 (10));
```

# **Контрольные вопросы и задания для самостоятельного выполнения**

1. Найдите ошибку в скрипте создания таблицы.

```
CREATE TABLE ORACLE1 (81NAME  
VARCHAR2 (20), ITEMS NUMBER);
```



# Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.